DBCASE 2.0

Miguel Arriba García

UNIVERSIDAD COMPLUTENSE
MADRID

Facultad de informática

Trabajo de fin de grado del Grado en Ingeniería Informática

Curso 2018/2019

Director: Fernando Sáenz Pérez

Índice general

Agradecimientos

A Fernando por darme la posibilidad de continuar este proyecto y por ayudarme con todas las dudas y mostrarse accesible en todo momento.

A David por la platilla y por las clases de LaTeX con las que he podido redactar esta memoria.

Y a mi familia y amigos por apoyarme siempre y por tener paciencia conmigo durante todo este tiempo.

Capítulo 1

Resumen

DBCASE 2.0 ofrece una manera fácil de construir bases de datos relacionales a partir del diseño de un diagrama entidad-relación, la herramienta permite generar al usuario el modelo lógico y físico a partir del diagrama, así como ejecutar directamente el código generado en un gestor de bases de datos.

El principal objetivo de la herramienta es **guiar** al usuario **en el proceso de diseño** de una base de datos relacional pasando por todas sus fases. La herramienta está pensada para que los usuarios experimenten con distintos diseños de forma que les permita adquirir y asentar conocimientos.

DBCASE está pensada desde un principio como un programa ligero y **multiplataforma**, lo que permite una gran versatilidad para ser utilizada en cualquier tipo de entorno.

A partir de un diagrama entidad-relación creado por un usuario, la aplicación es capaz de generar el modelo lógico del diseño, así como el modelo físico adaptado a tres gestores de bases de datos distintos: MySQL, ORACLE y MS ACCESS.

La aplicación ofrece una solución completa a todas las fases de diseño de una base de datos, además de estar específicamente diseñada en el proceso de aprendizaje, lo cual la diferencia de las alternativas existentes en el mercado.

1.1. Palabras Clave

Bases de datos | Java | Diagrama Entidad Relación | Modelo Relacional | SQL

Capítulo 2

Abstract

DBCASE 2.0 offers an easy way to build relational databases from the design of an entity-relationship diagram, the tool allows the user to generate the logical and physical model from the diagram, as well as directly execute the code generated in a DBMS.

The main goal of the tool is to guide the user in the process of designing a relational database through all its phases. The tool is designed for users to experiment with different designs in a way that allows them to acquire and establish knowledge.

DBCASE is designed from the beginning as a lightweight and cross-platform program, which allows great versatility to be used in any type of environment.

From an entity-relationship diagram created by a user, the application is capable of generating its logical model, as well as the physical model adapted to three different database managers: MySQL, ORACLE and MS ACCESS.

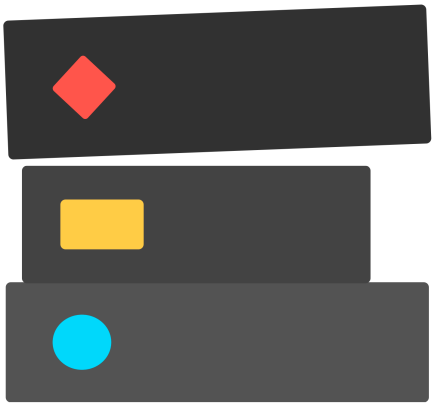
The application provides a complete solution for all phases of designing a database, as well as being specifically designed for the learning process, which differentiates it from the existing alternatives in the market.

2.1. Keywords

Databases | Java | Entity Relationship Diagram | Relational Model | SQL

Capítulo 3

DBCASE 2.0. Introducción



DBCASE 2.0 surge como una actualización al proyecto DBCASE con la que se pretende dar un nuevo aspecto al diseño de la interfaz de usuario y continuar con la implementación de nuevas funcionalidades.

El programa está destinado principalmente al uso académico y por ello ha sido pensado como una herramienta didáctica que ayude al alumno en su proceso de aprendizaje, aunque también puede ser usada profesionalmente como una manera fácil y rápida de construir una base de datos relacional. Por ello la herramienta permite al usuario crear una base de datos funcional a partir de un diagrama sin necesidad de que conozca ningún lenguaje de programación de base de datos, lo que es perfecto para alumnos que se estén iniciando en la materia.

La herramienta está desarrollada utilizado el lenguaje de programación Java. Tam-

bién cuenta con elementos creados mediante html y css.

3.1. Motivación

Dado que el uso de las bases de datos relacionales está ampliamente extendido, desde hace años existen herramientas pensadas para la creación de una base de datos a partir del diseño de un diagrama, simplificando el proceso.

DBCase pretende centrarse en el proceso de aprendizaje del diseño y creación de una bases de datos, llevando al usuario a través de todas las fases e informándole de los errores que puede tener su diseño. Por ello DBCase está pensada especialmente para ser utilizada en un ámbito académico diferenciándola así de otras alternativas disponibles.

3.2. Antecedentes

El proyecto DBCase 2.0 es la continuación de dos proyectos anteriores.

DBDT de sistemas informáticos (2007/2008)

El proyecto fue el punto de partida de la aplicación. Pretendía dar una alternativa al diseño de bases de datos en papel, enfocándose en el ámbito académico. Se realizó un programa multiplataforma optando por el lenguaje java.

Autores del proyecto:

- Alberto Milán Gutierrez, Miguel Martinez Segura, Francisco Javier Cáceres González
- Profesora Directora: Yolanda García Ruiz

DBCase de sistemas informáticos (2008/2009)

Este proyecto supuso una ampliación y mejora de las funcionalidades que ofrecía el proyecto del año anterior, se mejoró el diseño del diagrama entidad relación y se creo la posibilidad de traducir el diagrama a código soportado por distintos gestores de bases de datos.

Autores del proyecto:

- Rodrigo Denis Cepeda Mateos, Cristina Marco de Francisco, Tello Serrano Gordillo

El proyecto servía como una herramienta bastante útil, pero que debido al paso de los años había quedado algo obsoleta. Sobre todo en lo que respecta a su interfaz gráfica.

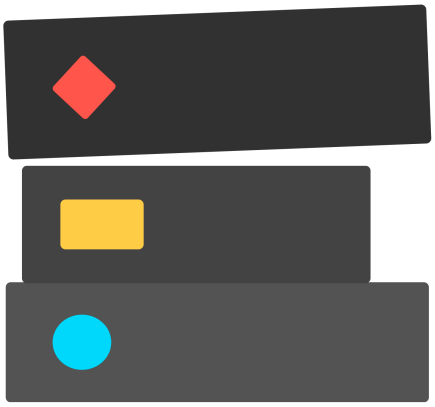
3.3. Objetivos

El principal objetivo del proyecto era rehabilitar la aplicación para su uso académico en las aulas, para ello se establecieron los siguientes objetivos.

- Actualizar el diseño de la interfaz gráfica de la aplicación, haciéndola más usable y moderna.
  - Cambiar la disposición de los paneles, de manera que el flujo de trabajo de la aplicación resulte más simple e intuitivo.
  - Rediseñar el panel de códigos, separando el modelo lógico del físico y simplificando el reporte de errores y advertencias.
  - Mejorar la representación del diagrama entidad relación.
- Corregir los posibles errores que arrastrase la aplicación desde versiones anteriores y continuar añadiendo funcionalidades nuevas.
  - Completar la traducción al modelo lógico, añadiendo indicaciones para que el usuario entienda cómo se ha realizado.
  - Añadir la posibilidad de escoger un tema oscuro para la interfaz gráfica.

Capítulo 4

DBCASE 2.0. Introduction



DBCASE 2.0 emerges as an update to the DBCASE project, which aims to give a new look to the design to the user interface and continue with the implementation of new functionalities.

The program is primarily intended for academic use and therefore has been thought of as a teaching tool that helps students in their learning process, although it can also be used professionally as an easy and fast way to build a relational database. Therefore, the tool allows the user to create a functional database from a diagram without needing to know any database programming language, which is perfect for students who are starting in the subject.

The tool is developed using the Java programming language. It also has elements created using html and css.

4.1. Motivation

Since the use of relational databases is widespread, for years there have been tools designed to create a database from the design of a diagram, simplifying the process.

DBCase aims to focus on the process of learning the design and creation of a database, taking the user through all phases and informing him of the errors that his design may have. Therefore, DBCase is specially designed to be used in an academic context, differentiating it from other available alternatives.

4.2. Previous projects

The DBCase 2.0 project is the continuation of two previous projects.

DBDT de sistemas informáticos (2007/2008)

The project was the starting point of the application. It was intended to give an alternative to the design of paper databases and was specially designed for use in classrooms. The program was multiplatform and made in Java language.

Project authors:

- Alberto Milán Gutierrez, Miguel Martinez Segura, Francisco Javier Cáceres González
- Director Professor: Yolanda García Ruiz

DBCase de sistemas informáticos (2008/2009)

This project meant an extension and improvement of the functionalities offered by the project of the previous year, the design of the relationship entity diagram was improved and the possibility of translating the diagram into code supported by different database managers was created.

Autores del proyecto:

- Rodrigo Denis Cepeda Mateos, Cristina Marco de Francisco, Tello Serrano Gordillo

The project served as a fairly useful tool, but that due to the passing of the years had become somewhat obsolete. Especially in regards to its graphical interface.

4.3. Goals

The main objective of the project was to rehabilitate the application for academic use in the classroom, for this purpose the following objectives were established.

- Update the design of the graphical interface of the application, making it more modern and usable.
  - Change the layout of the panels, making application workflow simpler and more intuitive.
  - Redesign the code panel, separating the logical model from the physical and simplifying the reporting of errors and warnings.
  - Improve the representation of the entity relationship diagram.
- Correct any errors that dragged the application from previous versions and continue to add new features.
  - Complete the translation into logical model, adding instructions for the user to understand how it was done.
  - Add the possibility to choose a dark theme for the graphic interface.

Capítulo 5

Estudio de soluciones existentes

Durante la primera fase de este proyecto se realizó una investigación y análisis de herramientas disponibles en el mercado que tuvieran características y funcionalidades similares a las que ofrece DBCase.

En este proceso de estudio se han investigado herramientas que permiten la realización de diagramas entidad relación, analizando las mecánicas que ofrecen para la creación de los mismos.

También se han investigado herramientas que permitieran la traducción de un esquema conceptual a un código ejecutable por un gestor de base de datos, analizando el flujo de trabajo llevado a cabo por la herramienta.

5.1. Draw.io

Draw.io [1] es una aplicación web de código abierto que permite realizar de manera sencilla todo tipo de diagramas siendo una de las herramientas más usadas a nivel mundial para este cometido.

Draw.io proporciona una gran cantidad de distintos elementos y formas que pueden ser usadas en el diagrama y proporciona una gran versatilidad y capacidad de personalización sin dejar de ser una herramienta muy intuitiva.

Con respecto al proyecto DBCase la herramienta ofrece soluciones muy interesantes en cuanto a la fase de diseño del diagrama entidad relación.

- Dispone de una barra lateral de elementos y una barra superior de herramientas, mostrando al usuario todos los elementos disponibles así como todas las acciones realizables por el usuario en un solo vistazo.
- Permite guardar y abrir proyectos, tanto desde el sistema de ficheros como desde otros sistemas de almacenamientos en la nube.
- Permite tener varios proyectos abiertos al mismo tiempo.
- El establecimiento de conexiones entre elementos se puede realizar de forma gráfica.
- Ofrece la posibilidad de cambiar el tema de la interfaz.

5.2. Smart Draw

Smart Draw [2] es una herramienta de software privativo que permite realizar múltiples tipos de diagrama. Cuenta con una versión web y otra de escritorio para los principales sistemas operativos. Está enfocada principalmente en diagramas de ingeniería y arquitectura y cuenta con un módulo específico para la creación de diagramas entidad relación muy completo.

En cuanto al espacio de trabajo que proporciona es muy similar al ofrecido por la herramienta Draw.io (posee una barra lateral de elementos y otra superior de herramientas, permite guardar y abrir proyectos y trabajar con varios al mismo tiempo).

Otro punto a destacar es la facilidad de conectar y crear elementos en el diagrama. Llevando el cursor al borde de un elemento permite crear otro idéntico directamente conectado. De esta forma se simplifica enormemente el proceso de conexión de elementos. Esta función es interesante y podría extrapolarse para añadir nuevas entidades a una relación. Esta funcionalidad es complementaria a la de establecer conexiones de forma gráfica, que también ofrece Draw.io.

5.3. EDRPLus

EDRPlus [3] es una aplicación web de uso gratuito, que está específicamente diseñada para la creación de diagramas de bases de datos. La herramienta da la opción de crear diagramas entidad relación y esquemas relacionales. Además la aplicación

permite traducir el diagrama entidad relación directamente a código sql.

La herramienta permite guardar proyectos y, mediante el inicio de sesión subir y acceder a tus proyectos guardados en el servidor, sacando un gran provecho al hecho de ser una aplicación web.

Como vemos, la herramienta permite realizar las dos funcionalidades principales que pretende DBCase.

1. En cuanto al espacio de trabajo proporcionado para el diseño del diagrama entidad relación, la herramienta es, en general menos potente e intuitiva que las vistas anteriormente. La creación de elementos y conexiones se realiza mediante un panel superior en el que aparecen todos los elementos disponibles (atributos, entidades, relaciones y etiquetas), y todas las acciones disponibles (establecer una conexión, seleccionar, eliminar, deshacer y rehacer) en una sola fila de botones de texto, lo que resulta confuso.

   Al tener un elemento seleccionado aparece un panel de opciones en la parte derecha de la pantalla que permite modificar todas las opciones del mismo (nombre, tipo, cardinalidad en el caso de relaciones etc). Esto parece una buena solución aunque puede dar pie a que se queden elementos sin modificar, ya que en ningún momento obliga al usuario ni siquiera a establecer un nombre para el elemento.
2. La herramienta no comprueba en ningún momento la corrección del diagrama. Puedes crear atributos sueltos o entidades débiles sin entidad fuerte, en este sentido es menos potente de lo que ya ofrece DBCase.

3. Para generar el código SQL, el usuario debe salir del panel de edición e ir al listado de sus documentos, solo desde ahí y mediante un menú desplegable se da la posibilidad de generarlo, lo cual resulta muy poco intuitivo.
4. No dispone de atajos de teclado.

5.4. SqlDBM

SqlDBM [4] es una herramienta web de software privativo que permite realizar el diseño de una base de datos mediante su diagrama relacional y traducirlo a código

ejecutable.

Se trata de una herramienta muy profesional y completa y está pensada para la creación de bases de datos de forma rápida e intuitiva, sin necesidad de conocer el lenguaje de definición de datos.

Tras un análisis, se han podido extraer las siguientes características acerca de la aplicación.

- Permite realizar un diagrama relacional de tablas, pero no permite realizar el diagrama entidad relación.
- Soporta varios gestores de bases de datos.
- Permite realizar la traducción de diagrama a código y viceversa, utilizando ingeniería inversa. Para ello dispone de dos botones muy visibles en una barra lateral.
- El área de trabajo cuenta con una barra de herramientas superior representadas mediante iconos. También permite añadir elementos (que en este caso son tablas, conexiones y notas).
- Al añadir una tabla se obliga al usuario a rellenar directamente su nombre y sus atributos. Una vez creada da la posibilidad de modificarla pulsando sobre ella.
- Posee un panel de información en el que muestra en forma de lista todas las tablas y conexiones existentes en el esquema.
- Permite abrir y guardar proyectos, así como trabajar sobre ellos de forma colaborativa.
- Posee un tema oscuro y otro claro para la interfaz.

5.5. Conclusiones del estudio

La finalidad del estudio ha sido analizar las alternativas disponibles en el mercado observando cómo dan solución a los distintos problemas que se plantean al realizar una aplicación de este tipo.

	Diagrama ER	Traducción a modelo relacional	Traducción a código BD
Draw.io	Sí	No	No
Smart Draw	Sí	No	No
EDRPlus	Sí	Diagrama	Sí
SqlDBM	No	Diagrama a partir de código	Sí
DBCase	Sí	En formato texto	Sí

En esta tabla se realiza una comparación entre los 4 programas analizados y DBCase, contrastando si dan solución o no a las principales funcionalidades que ofrece ya nuestra herramienta.

- **Draw.io** y **Smart draw** son dos herramientas muy similares que ofrecen una solución muy buena a la hora de realizar el diseño del diagrama entidad relación. En este aspecto, pueden servir como referencia para mejorar la experiencia de diseño ofrecida por DBCase.
- **SqlDBM** es de las herramientas analizadas la más enfocada a un ámbito profesional. Aunque no ofrece la posibilidad de crear un diagrama entidad relación, cubre de forma excelente el resto de funcionalidades que ofrece DBCase.
- **EDRPlus** es quizás la herramienta más similar a DBCase, ya que cubre las tres funcionalidades principales ofrecidas por la herramienta.
  - EDRPlus permite realizar un diseño del diagrama entidad relación muy similar al que puede realizarse con DBCase, sin embargo la notación usada para las relaciones no es la misma que la estudiada en la facultad.

    También da una mayor libertad al usuario a la hora de crear nuevos elementos, lo cual en este tipo de diagramas puede ser contraproducente ya que permite que se cometan más errores de diseño.
  - Permite traducir el diagrama entidad relación a un diagrama relacional, pero no proporciona una alternativa en formato texto.
  - El flujo de trabajo en EDRPlus es en general poco intuitivo, y para realizar las traducciones es necesario rebuscar entre los menús.
  - Como punto a favor, EDRPlus es una aplicación web, no requiere instalación y permite almacenar los proyectos en su propio servidor.

De forma general, a raíz del estudio se puede decir que ninguna de las herramientas analizadas alcanza por completo los objetivos que pretende la herramienta DBCase,

aunque se pueden extraer ideas muy útiles de como las mismas han solucionado los distintos problemas de diseño.

Capítulo 6

Memoria de trabajo

Mi trabajo se ha centrado en mejorar la herramienta en los aspectos gráficos y funcionales, y añadiendo en la medida de lo posible todas las funcionalidades que hemos considerado necesarias para la aplicación.

A continuación se describen con detalles los principales cambios y mejoras que se han implementado con respecto a la versión anterior.

6.1. Diseño

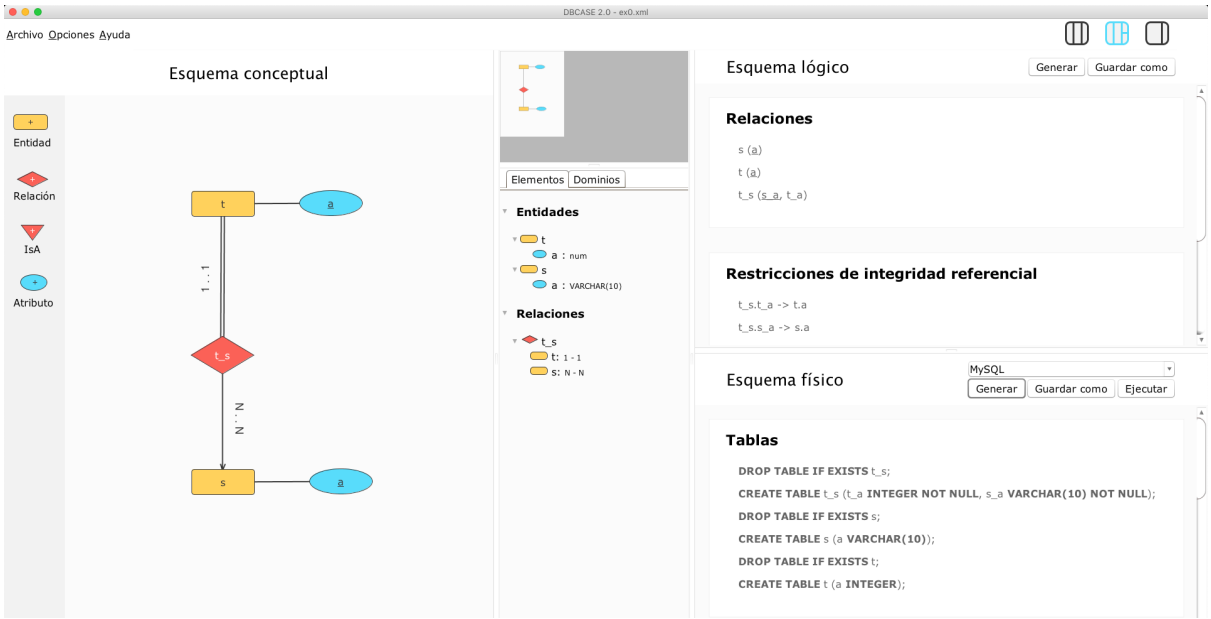


Figura 6.1: GUI de DBCASE 2.0

La aplicación ha sufrido un profundo cambio de diseño. Se ha buscado dar un aspecto más moderno a la aplicación, de forma que se han conservado las librerías gráficas con las que implementó y se ha intentado exprimir al máximo sus posibilidades.

6.1.1. Look and Feel

Debido a la decisión de continuar usando las librerías de java swing, se investigaron las posibilidades que ofrecía para personalizar el diseño de la interfaz. Una de las principales opciones que ofrece java swing es la de poder editar fácilmente el estilo de los paneles mediante el denominado look and feel.

```
1 javax.swing.UIManager.setLookAndFeel();
```

El look and feel en java swing permite cambiar el aspecto por defecto de todos los componentes visuales generados por la librería (JPanel, JFileChooser, JFrame, etc).

Nimbus Look and Feel

Java ofrece por defecto varios look and feel con los que el programador puede editar sus interfaces.

De las distintas opciones se decidió utilizar Nimbus Look and Feel.

> *"Nimbus es un pulido Look and Feel multiplataforma introducido en Java SE 6 Update 10 (6u10)."*[5]

Entre los principales motivos que llevaron a tomar esta decisión destacan:

- Su aspecto moderno y redondeado suponía un importante lavado de cara con respecto al aspecto anterior, y era una cualidad diferencial con respecto a las otras opciones.
- El hecho de que sea multiplataforma permite que la aplicación tenga un aspecto uniforme en los distintos sistemas operativos.
- La capacidad de personalización, que permite elegir el aspecto de prácticamente cualquier elemento visual de la interfaz.

Nimbus Look and Feel personalizado

La personalización de Nimbus se ha centrado principalmente en dos aspectos.

- La fuente de texto: se ha buscado que la aplicación sea lo más accesible y clara posible, por lo que se ha utilizado la fuente **Verdana 16**, por ser una de las más usadas y simples, y a tamaño 16 para que la lectura por parte del usuario se realice sin dificultades.

- Los colores de los elementos, paneles, botones y menús. De forma que permitiera la implementación de distintos temas.

6.1.2. Temas

Un aspecto en el que se ha centrado el trabajo ha sido en la implementación de temas que permitieran al usuario personalizar la interfaz gráfica.

Los dos objetivos que se perseguían con la implementación de temas eran:

- Habilitar un tema oscuro, tan usado en las interfaces de usuario modernas, que permite reducir la fatiga visual y hace más cómodo el trabajo en entornos con poca luz.
- Permitir a los usuarios crear y personalizar sus propios temas, pudiendo escoger los colores principales que utiliza la aplicación.

Detalles de la implementación de los temas

Para que la creación y personalización de temas fuera lo más dinámica posible se decidió que los distintos temas se almacenasen en archivos **json**[6], externos al programa, de forma que el usuario pueda modificarlos, copiarlos o eliminarlos fácilmente. Los archivos json que contienen los temas se almacenan en la carpeta "themes".

> *"JSON (acrónimo de JavaScript Object Notation) es un formato de texto sencillo para el intercambio de datos."*

Dentro del archivo json, se establecen el nombre del tema, y los colores de los principales elementos de la aplicación, entre ellos algunos de los más destacables son:

- El color de los elementos del diagrama (relaciones, atributos y entidades).
- El color de fondo del diagrama.
- El color principal de la aplicación (utilizado por Nimbus para pintar el fondo de los paneles, botones, etc).
- El color de las fuentes, tanto de la fuente de los paneles como de la fuente de los cuadros de texto de los esquemas relacional y físico.

Dentro del menú de opciones, el usuario tiene la posibilidad de elegir un tema de entre todos los temas disponibles dentro de la carpeta "themes".

Se han incorporado dos temas con la aplicación:

- **Light** o tema claro: es el tema por defecto. Utiliza colores blancos y grises para la aplicación, y colores elementales para resaltar los elementos del diagrama.
- **Dark** o tema oscuro: tiñe la interfaz con colores negros manteniendo el protagonismo de los elementos del diagrama.

Además el usuario puede crear sus propios temas añadiendo un archivo en formato json a la carpeta "themes"que siga la estructura de los temas incorporados.

Para el manejo interno de los temas se ha creado una clase theme utilizando el patrón de diseño **"Singleton"**[7], lo que permite a cualquier otra clase obtener una instancia de la misma y poder consultar en ella los colores de forma sencilla.

6.1.3. Disposición de paneles

Uno de los mayores cambios que ha sufrido la interfaz de usuario ha sido la disposición de los distintos paneles. Se ha buscado una GUI más clara y sencilla sin perder ninguna de las funcionalidades.

Con respecto a la versión anterior, los cambios principales que ha sufrido la interfaz son:

- Se ha pasado de una interfaz dividida en dos pestañas (diagrama entidad-relación y generación de código) a una sola, de forma que ahora se puede ver de un vistazo el diagrama y el código generado.
- El panel de generación de código se ha dividido en dos paneles distintos: Esquema lógico y esquema físico. De esta manera quedan ambos lenguajes en paneles independientes.
- Se han unificado el panel de información y el panel de dominios en un solo JTabbedPane.
- Los paneles están separados mediante la clase JSplitPane [8], lo que permite al usuario cambiar a su antojo el tamaño de cada panel.
- Se ha eliminado el panel de sucesos, ya que su utilidad era más bien para el desarrollo de la aplicación y no para el usuario.

6.1.4. Panel de información

Figura 6.2: Panel de Información

El panel de información muestra un esquema en forma de árbol del modelo entidad relación existente en el panel de diseño, y en él se puede ver de forma esquemática todos los elementos y cómo están relacionados.

El panel de información ha sufrido profundos cambios que se describen a continuación:

- Ahora el panel muestra en todo momento la información global de todo el esquema creado por el usuario, al contrario que en la versión anterior, en la que solo mostraba la información de los elementos seleccionados.
- El panel de información ha pasado a dividirse en dos sectores: entidades y relaciones.
  - **Entidades**: muestra todas las entidades del esquema y sus atributos. Muestra el dominio del atributo al lado del nombre y representa gráficamente de forma correcta los subatributos de atributos compuestos (como ramas de los mismos).
  - **Relaciones**: muestra todas las relaciones del esquema y para cada una de ellas muestra sus atributos y las entidades que relaciona. Para cada una de las entidades muestra la cardinalidad al lado del nombre.

Para las relaciones ISA muestra la entidad padre como la primera entidad de la relación, y las entidades hijas con un icono que las diferencia como tal.

- Para la renderización del árbol (implementado con la clase JTree), se ha creado una clase personalizada que hereda de la clase DefaultTreeCellRenderer. Con ella se ha podido personalizar los iconos y las fuentes de cada uno de los elementos del árbol en función de su tipo, mediante la sobreescritura del método getTreeCellRendererComponent().

  Los distintos iconos son generados mediante la clase Graphics2D [9], de forma que se evita el uso de imágenes externas pregeneradas, y permite que los colores de los iconos se adapten al tema seleccionado.

6.1.5. Panel de dominios

El panel de dominios informa al usuario de las opciones disponibles que tiene el usuario para asociar con los atributos creados. En este panel se han realizado los siguientes cambios.

- El panel tiene un mayor tamaño y ahora muestra todos los dominios disponibles en el sistema, no solo los creados por el usuario.

- Solo para los dominios creados manualmente se muestra el conjunto de elementos disponibles y el tipo base al que pertenece el dominio, manteniendo la capacidad de editarlos.

- Se ha creado un nuevo renderizador para representar gráficamente el árbol, eliminando los iconos de ficheros de la versión anterior y dejando una interfaz más limpia e intuitiva. El texto presente en el panel se crea utilizando código HTML, y en función de la jerarquía que tenga dentro del árbol se aplica uno u otro formato.

- Se ha habilitado un botón para añadir nuevos dominios al sistema de forma sencilla.

6.1.6. Barra lateral para añadir elementos

Figura 6.3: Barra lateral para añadir elementos

Dentro del área de diseño se ha añadido una barra lateral que permite agregar nuevos elementos al diagrama. De esta forma se consigue que el usuario pueda crear elementos nuevos de una forma más intuitiva que haciendo clic derecho sobre el panel de diagrama. A continuación se detallan las principales características de este panel.

- Permite al usuario añadir cuatro tipos de elementos: entidades, relaciones, relaciones ISA y atributos.

  Los botones de añadir entidad y añadir relación muestran el cuadro de diálogo correspondiente, en el que se permite insertar el nombre y las características del elemento.

  El botón ISA, al no necesitar nombre ni configuración, inserta directamente la relación en el diagrama.

  El botón atributo permite seleccionar mediante un menú desplegable a qué entidad o relación pertenece el atributo, también se le pide al usuario el nombre y las características del atributo.

- Los elementos se insertan automáticamente en el diagrama. Al no haber hecho clic derecho el usuario en ningún punto del diagrama, el programa no dispone de una posición de referencia en la que el elemento debe ser insertado. Por ello el programa genera unas coordenadas automáticamente.

  - Las **entidades** y **relaciones** nuevas empiezan a colocarse a partir de la esquina superior izquierda del panel. Los elementos sucesivos se colocan a la derecha del anterior, dejando un pequeño margen.

    El programa dispone del tamaño exacto que tiene el panel de diseño en cada momento, por tanto cuando se intenta insertar un elemento a la derecha del anterior, pero quedando fuera del diagrama, el programa lo genera en una línea inferior.

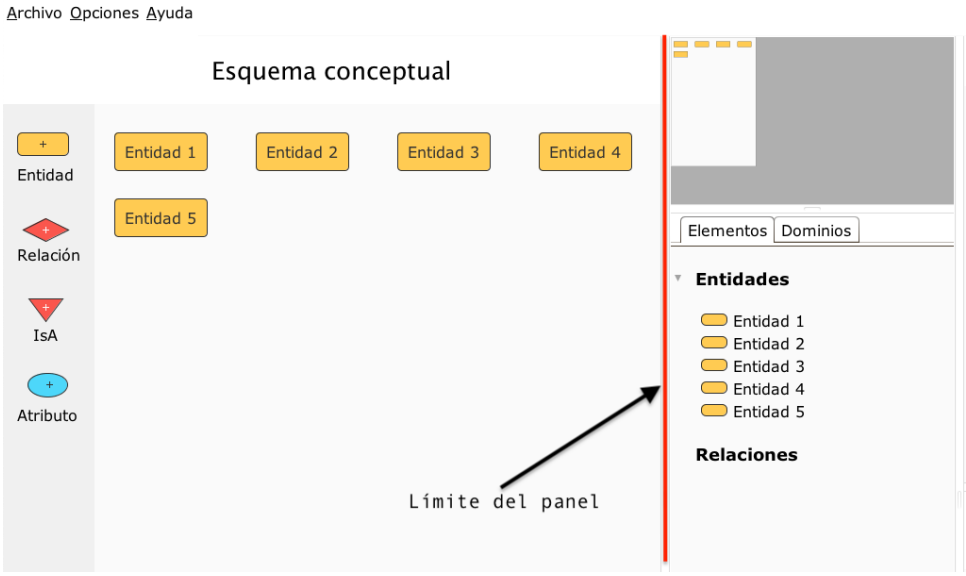


Figura 6.4: Colocación de nuevos elementos

  - Los **atributos** siguen un comportamiento distinto debido a que dependen de la relación o entidad a la que pertenezcan. Los nuevos atributos se generan creando círculos concéntricos alrededor del elemento al que pertenecen, a diferencia de la versión anterior en la que simplemente se generaban a la derecha del elemento.

    Para realizar el cálculo de las coordenadas se ha creado una función con la ayuda de la función "sin"de la librería Math [10], que tiene en cuenta el número de atributos del elemento, el ancho del elemento y la distancia a los márgenes

del diagrama (evitando que se generen nuevos atributos en posiciones negativas, obligando al usuario a desplazar el diagrama).

```
p.setLocation(
    Math.round(ancho*Math.sin(offset/(Math.PI/constant))+p.getX()),
    Math.round(alto*Math.sin(offset/(Math.PI/constant)-(Math.PI/2))+p.getY())
);

```

6.1.7. Paneles de texto

La forma en la que se presentan los códigos generados ha cambiado radicalmente, como se ha comentado anteriormente se ha dividido en dos paneles (modelo lógico y modelo físico). A continuación se describen en detalle todos los cambios que han sufrido.

- Se han eliminado los botones de validar modelo y de limpiar área de texto.
- La validación del modelo ahora se hace de forma automática cuando el usuario pulsa el botón de generar código (tanto para el modelo lógico como para el modelo físico).
- La validación ahora solo muestra mensajes cuando existen errores o alertas en el diagrama. Anteriormente informaba de todas las comprobaciones realizadas, fueran correctas o no.
- Ha desaparecido el botón de limpiar área y en su lugar los cuadros de texto son completamente editables, dando libertad al usuario para modificar o añadir cualquier texto que desee.
- Los paneles han sufrido también un gran cambio en el aspecto visual. Están construidos mediante una clase ReportPanel"que hereda de JTextPane. La clase modifica el panel pana que muestre contenido HTML mediante la función.

```
setContentType("text/html");
```

El texto ahora se interpreta como código HTML y se representa como tal, lo que amplía enormemente las posibilidades de diseño. Se ha añadido en el código una sección css con los estilos del panel. Los estilos son generados teniendo en cuenta el tema seleccionado por el usuario y modifican varios aspectos como:

- Párrafos.

- Títulos.
- Listas.
- Divisores.
- Color de fondo.
- Fuente y color de las letras.

Figura 6.5: Ejemplo de modelo relacional.

Como se puede observar se ha optado por un diseño que divide en tarjetas las distintas secciones del código. Los mensajes de error y de aviso aparecen en distintos colores y se ubican siempre en la parte superior.

Figura 6.6: Ejemplo de mensajes de aviso y error.

6.1.8. Perspectivas

Se ha implementado la posibilidad de elegir distintas perspectivas en las que se colocan los paneles en función de las necesidades del usuario. En la parte superior derecha de la aplicación aparece un menú de selección de perspectiva. Los iconos son generados mediante la librería Graphics2D [9] y se adaptan a los colores del tema. Funcionan como "Botones de Radio", es decir que solo se permite tener seleccionada una de las perspectivas.

Figura 6.7: Menú de selección de perspectiva.

Se han desarrollado tres perspectivas:

- **Código**: Divide la pantalla en tres paneles: panel de información / dominios, panel de esquema lógico y panel de esquema físico. Está pensado para permitir al usuario centrarse en la edición de códigos. Mantiene el panel de información / dominios para servir de referencia del diagrama dibujado.
- **Completo**: Integra todos los paneles posibles y sirve para que el usuario tenga una visión global de todo el trabajo.
- **Diseño**: Muestra el panel de diseño, la barra de añadir elementos y el panel de información / dominios. Con él se centra la atención en el diseño del modelo entidad relación dando todo el espacio de la pantalla para ello.

6.1.9. Cambios en el panel de diseño

El panel de diseño también ha sufrido algunos cambios desde la versión anterior, aunque en general el panel sigue teniendo un comportamiento similar y los cambios producidos aquí han sido de poco impacto.

- Se ha cambiado el aspecto de los elementos, añadiendo colores, resaltando los bordes cuando el elemento está seleccionado, cambiando la fuente de texto, y redondeando los bordes de las entidades.
- Se ha acercado el zoom por defecto tanto en el panel principal como en en la miniatura, y se ha corregido la orientación del scroll para hacer zoom (funcionaba

al revés de lo intuitivo).

- Se han añadido flechas en las relaciones que lo requieren y también se han implementado las dobles líneas para las relaciones de participación total.

6.1.10. Cambios en los cuadros de diálogo

DBCase dispone de cuadros de diálogo para realizar tareas como crear una relación, editar la cardinalidad de una entidad o guardar el proyecto en un archivo.

En ésta versión se han modificado todos los cuadros de la siguiente manera.

- Se han adaptado todos los elementos para adaptarlos a los temas elegidos por el usuario.
- Se han eliminado los iconos que acompañaban a muchos de los cuadros y se ha simplificado su diseño. Se ha aumentado el tamaño de los cuadros de introducción de texto.

6.1.11. Menú

El menú de opciones también ha sufrido algunos pequeños cambios.

- Se han quitado los iconos que acompañaban a algunas de las opciones del menú, se ha aumentado el tamaño de letra y se ha conseguido que el menú se adapte al tema vigente. Para ello ha sido necesario crear una clase que herede de JMenuBar [11] y que modifique uno a uno los componentes necesarios.
- En el menú de opciones se han quitado las opciones de ocultar el panel de sucesos (el panel ya no existe), y seleccionar el gestor de Base de Datos actual (Esta opción se ha desplazado al panel de esquema físico).

  En su lugar se han añadido un menú desplegable para seleccionar el tema de entre todos los disponibles en la carpeta "themes", y una opción para mostrar o no los atributos posiblemente nulos.
- Se ha actualizado el texto de “Acerca de DBCASE”.

6.1.12. Archivo abierto

En la versión anterior existía un panel al pie de la aplicación que indicaba el directorio de trabajo actual, con la información del archivo temporal que se estaba usando para

guardar los cambios en la aplicación. Esta información no era demasiado relevante para el usuario, y sería más útil indicar cual es el archivo abierto por el usuario.

También existía en la barra de menús un botón que funcionaba como un semáforo de cambios (adoptaba el color verde cuando todos los cambios estaban guardados y el color rojo cuando existían cambios en el archivo temporal sin guardar en el archivo final). Finalmente este botón se ha quitado.

Como solución se ha pensado indicar en la cabecera del programa cual es el archivo (xml) abierto por el usuario (en caso de que no haya creado un archivo nuevo). Si existen cambios en el programa sin guardar, el nombre del archivo viene seguido de un asterisco (*).

De esta forma se han unificado y mejorado ambas funcionalidades en un espacio más reducido.

6.1.13. Logotipo

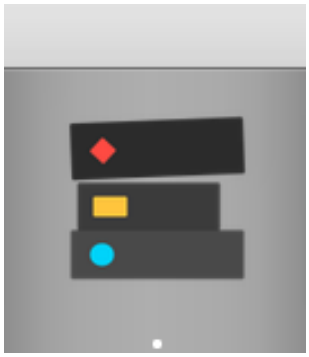


Figura 6.8: DBCASE en el dock de MacOS

Se ha diseñado un nuevo logotipo para la aplicación. El diseño está inspirado en el stack con el que se suelen representar las bases de datos. Se ha buscado darle aspecto de letra E mayúscula, ya que es la última letra del nombre del programa, y se han incluido una representación de los tres elementos principales del modelo entidad relación (atributos, entidades y relaciones).

El logotipo aparece como icono de la aplicación tanto en sistemas operativo Windows como MacOS.

6.2. Funcionalidad

Los cambios a nivel de funcionalidad se han centrado principalmente en las traducciones desde el esquema conceptual a los esquemas lógico y físico. Se han corregido algunos errores arrastrados desde la versión anterior, y se han implementado nuevas traducciones, mejorando sobre todo el esquema lógico.

6.2.1. Esquema lógico

La traducción del diagrama entidad relación al modelo relacional ha sufrido grandes mejoras en esta versión.

- Se ha depurado la traducción al modelo relacional, corrigiendo errores arrastrados desde la anterior versión.
- Se han añadido varias secciones nuevas al esquema lógico.
  - **Restricciones de integridad referencial**: En esta sección aparecen descritas de forma esquemática las restricciones de integridad referencial [12] generadas a partir del diagrama entidad relación.
  - **Restricciones perdidas**: Aquí aparecen todas las restricciones que pueden generarse a raíz del diagrama, pero que no son representadas en el modelo relacional.
    - **Cardinalidad**: indica las restricciones que se pierden con respecto a la cardinalidad de las relaciones, por ejemplo para una cardinalidad de 1 a 23, indica que se ha perdido la cardinalidad máxima de 23 en la entidad.
    - **Restricciones de tabla**: muestra las restricciones de tabla introducidas manualmente por el usuario, así como algunas restricciones asociadas a los atributos (por ejemplo que el atributo sea de tipo unique).
    - **Claves candidatas**: informa de los atributos que funcionan como clave candidata de una determinada tabla.

6.2.2. Exportación de códigos

Para realizar la traducción a los distintos sistemas gestores de bases de datos, la herramienta cuenta con varias clases (una por cada gestor), que heredan de una clase padre “ConectorDBMS”. En cada una de ellas se especifican las peculiaridades sintácticas de cada gestor, manteniendo los métodos equivalentes en la clase padre.

Con respecto a la exportación de los códigos se han implementado las siguientes mejoras.

- Se han implementado dos botones de exportación: uno para el esquema lógico y otro para el esquema físico. De esta forma se pueden guardar los códigos de forma independiente. El esquema lógico puede ser guardado en formato .txt, mientras que el esquema físico puede ser guardado en formato .sql y en formato .txt.
- Se ha añadido la capacidad de guardar el texto modificado escrito por el usuario, de forma que al guardar el texto se guarda con las modificaciones existentes.
- Esto también ocurre a la hora de ejecutar el código sql del esquema físico. El código ejecutado es el código con las modificaciones del usuario.
- Para hacer posible el guardado y la ejecución de los códigos modificados, y dado que los paneles de texto funcionan con el formato HTML, ha sido necesaria la implementación de un sistema de traducción de texto HTML a texto plano.

6.2.3. Atributos posiblemente nulos

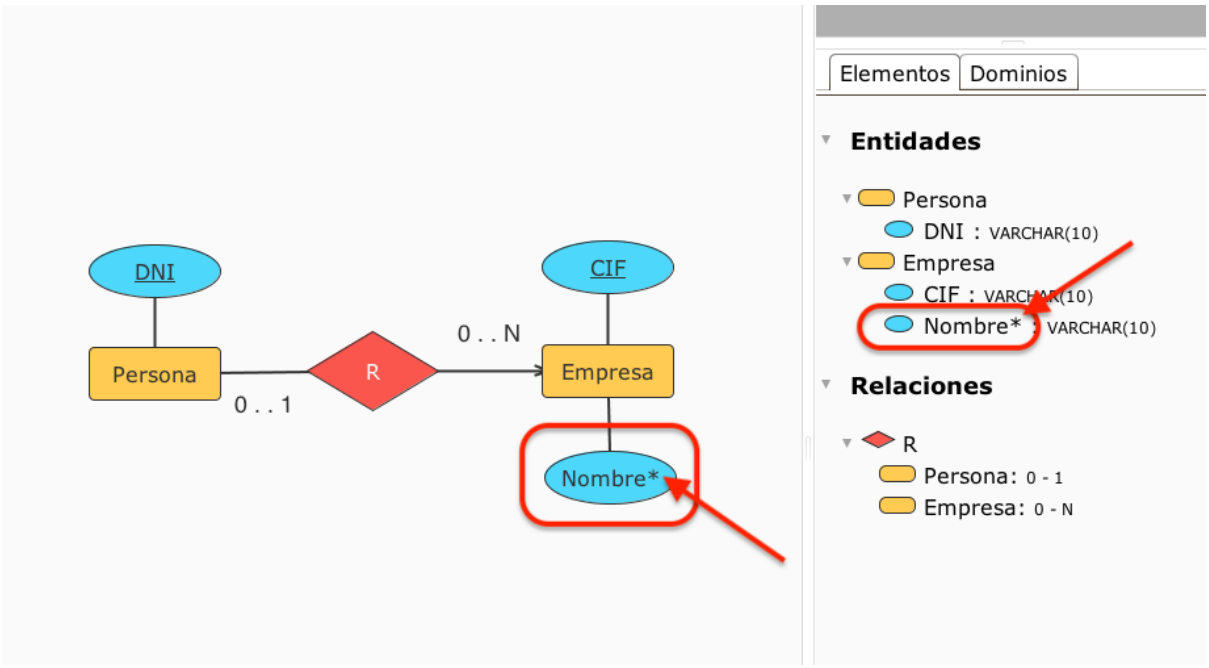


Figura 6.9: Representación de atributo posiblemente nulo.

Se ha implementado la capacidad de identificar los atributos posiblemente nulos. Si la opción está activada en el menú, los atributos que cumplan el requisito se mostrarán tanto en el panel de diseño, como en el panel de elementos y en el modelo relacional con un asterisco (*) al final del nombre del mismo.

6.2.4. Desplazamiento del diagrama

Se a añadido la opción de desplazar el diagrama en los ejes x e y manteniendo pulsada la barra espaciadora y arrastrando el panel. De esta forma se añade otra opción más a parte de la miniatura del diagrama para realizar el desplazamiento.

```
1  //Funcion que cambia el modo de uso del panel
2  public void toggleDragMode(boolean b) {
3      if(b==mouseMode) return;
4      if(!mouseMode) {
5      //cambia el panel a modo desplazamiento
6        creaGraphMouse();
7        //Se anade el plugin de desplazamiento
8        graphMouse.add(translating);
9        graphMouse.setMode(ModalGraphMouse.Mode.TRANSFORMING);
10       vv.setGraphMouse(graphMouse);
11     }else {//cambia el panel a modo edicion
12       creaGraphMouse();
13       //Se anade el plugin de edicion
14       graphMouse.add(picking);
15       graphMouse.setMode(ModalGraphMouse.Mode.PICKING);
16       vv.setGraphMouse(graphMouse);
17     }
18     mouseMode = !mouseMode;
19   }
```

El panel de diseño esta implementado usando la clase VisualizationViewer [13] (objeto vv en el ejemplo), disponible el la librería de Jung [14]. El funcionamiento del ratón viene dado por el objeto GraphMouse, cuyo funcionamiento depende del plugin que posea en cada caso.

6.2.5. Guardado de configuraciones

El archivo dbcase.config almacena algunas de las configuraciones del programa (como el lenguaje o el último archivo abierto), de forma que la próxima vez que el usuario inicie la aplicación estas configuraciones son recordadas. Ahora este archivo también almacena el último tema escogido por el usuario y la última perspectiva usada.

6.3. Código

6.3.1. Refactorización

El código de la aplicación ha sufrido un gran cambio durante todo el proceso de trabajo. Se ha intentado simplificar al máximo los paquetes y clases que configuran el

programa así como reducir en la medida de lo posible el código sobrante o redundante.

- Se han reordenado todas las clases a fin de estructurar el código según el patrón modelo vista controlador [15], de esta forma queda separada la parte gráfica de la aplicación de la parte lógica, quedando el controlador como intermediario entre las comunicaciones del usuario y el procesamiento de datos.

  En la anterior versión ya se disponía de una controlador, sin embargo las clases referentes a la vista y a la lógica de la aplicación estaban esparcidas por distintos paquetes de forma arbitraria. Ahora la carpeta src cuenta con cuatro paquetes: modelo, vista, controlador y persistencia (que contiene las clases necesarias para la lectura de archivos, utilizando el patrón DAO).

- Se ha intentado simplificar todo lo posible el código, haciéndolo más legible. Se ha procurado que las clases con muchas líneas de código se dividan en la medida de lo posible en otras clases que abarquen sus funcionalidades, de forma que el código quede mucho más mantenible. (Por ejemplo la clase controlador ha reducido su tamaño en 700 LDC, y la clase GUIPrincipal en 500 LDC, todo ello manteniendo intacto sus comportamientos).

- Muchos de los elementos de la vista fueron generados usando un software de creación de interfaces gráficas. Este tipo de herramientas, aunque son extremadamente potentes, el código que generan en muchas ocasiones es confuso y denso. Se ha intentado que el código que genera la vista quede más mantenible y legible, separando cada panel en su propia clase independiente.

- Se han eliminado todas las clases, paquetes y librerías que habían quedado en desuso. También se han eliminado todas las carpetas referentes al control de versiones mediante CVS.

- Con respecto al código usado para la parte gráfica de la aplicación, gran parte del mismo estaba ubicado dentro de una sola clase (la clase GUIPrincipal). Se ha modularizado la clase, sacando los elementos y paneles principales de la vista (como el menú o los paneles de códigos) y ubicándolos en sus propias clases. De esta forma resultan mucho más accesibles a la hora de realizar cualquier modificación.

- Se ha creado la clase Perspectiva, que maneja los paneles visibles en la aplicación.

Según la perspectiva seleccionada, la clase reorganiza todos los paneles y refresca la aplicación para mostrar los cambios.

```
1 // Metodo que establece la perspectiva de diseno.
2   public void modoDiseno() {
3     mainPanel.removeAll();
4     infoSplitMapa.add(infoPanel, JSplitPane.RIGHT);
5     mainPanel.add(diagrama);
6     infoSplitMapa.setResizeWeight(0.2);
7     diagramaSplitCode.setResizeWeight(0.1);
8     diagramaSplitCode.setVisible(false);
9     programmerSplit.setVisible(false);
10    mainPanel.revalidate();
11    mainPanel.repaint();
12    modo = 1;
13  }
```

- En cuanto a la generación de los códigos, en la anterior versión pulsar en el botón de generar código (tanto para el modelo lógico como para el físico), generaba un código HTML, que mostraba por pantalla, y otro en formato texto plano, que utilizaba en caso de que el usuario quisiera exportarlo.

  Ahora el código se genera únicamente en formato HTML, cuando el usuario desea exportar o ejecutar, éste código se traduce a texto plano mediante un método que elimina las etiquetas características del lenguaje. Con este cambio se ha simplificado enormemente el proceso de generación de código.

Nuevas carpetas

Ahora el proyecto incorpora dos nuevas carpetas.

- **Themes**: almacena los temas por defecto y los posibles temas creados por el usuario.
- **Projects**: Es la carpeta en la que se guardan por defecto todos los archivos generados por el usuario (proyectos o códigos sql y txt). Se crea automáticamente si no existe.

6.3.2. Repositorio

Anteriormente el proyecto utilizaba CVS [16] para el control de versiones. Durante este curso se ha decidido abandonar este método y pasar a utilizar un repositorio git.

Los cambios realizados durante todo el año están reflejados en un repositorio público [17] en Github.

A pesar de tratarse de un proyecto individual, el uso de un repositorio es útil para el control de versiones y como copia de seguridad del código.

Capítulo 7

Resultados, conclusiones y trabajo futuro

7.1. Resultados

DBCASE permite realizar el proceso de diseño y creación de una base de datos de forma sencilla mediante una interfaz de escritorio, centrándose en ayudar al estudiante en su proceso de aprendizaje.

Espacio de trabajo

El usuario puede modificar muchos aspectos del espacio de trabajo para hacer más cómoda su experiencia con la aplicación.

Puede escoger el idioma, el tema y la perspectiva que desee. También puede redimensionar los paneles para adaptarlos a sus necesidades.

Creación de un diagrama

El primer paso es la creación y el diseño de un diagrama entidad relación. Para ello se debe seleccionar una de las dos perspectivas que muestran el panel de diseño.

Para añadir elementos al diagrama el usuario dispone de dos opciones.

- Haciendo clic derecho sobre el panel de diseño. Se mostrará un menú desplegable que permite insertar cualquier tipo de elemento.
- Utilizando la barra lateral, pulsando en los iconos que representan los elementos.

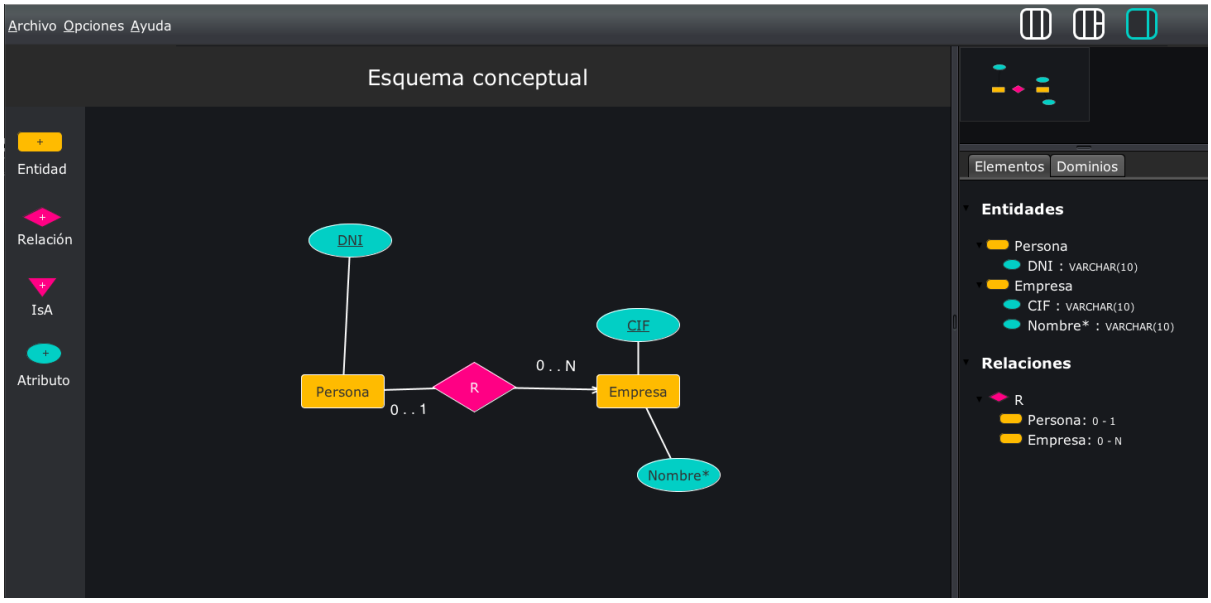


Figura 7.1: Ejemplo de espacio de trabajo personalizado

A la hora de insertar elementos, se pueden escoger mediante distintos cuadros de diálogo las características de los mismos. Por ejemplo en el caso de insertar una nueva entidad se debe indicar el nombre, y se da la opción de que la entidad sea débil con respecto a otra existente en el diagrama. En tal caso se debe seleccionar esa entidad fuerte y dar nombre a la relación que las une.

El procedimiento para eliminar elementos del diagrama es el siguiente:

- Primero se debe seleccionar un elemento o un conjunto de elementos (bien abarcando todos los elementos con el ratón o bien seleccionando de uno en uno y manteniendo pulsada la tecla shift).
- Tras realizar la selección se puede eliminar los elementos de dos formas:
  - Hacer clic derecho sobre el elemento, tras lo cual se mostrará un menú desplegable, en el que una de las opciones es la de eliminar el elemento.
  - Pulsando la tecla de suprimir.

  Antes de finalizar el proceso se mostrará un cuadro de confirmación.

Relaciones

Uno de los elementos básicos en los diagramas entidad relación son las relaciones entre entidades.

Para conectar una relación con una entidad el usuario debe hacer clic derecho sobre la relación y pulsar en la opción "añadir una entidad" del menú desplegable. Tras ello se abrirá un cuadro de diálogo en el que se debe seleccionar la entidad y elegir con qué cardinalidad se desea establecer el vínculo. También se da la posibilidad de indicar un nombre para el rol del enlace.

Restricciones

El usuario puede añadir restricciones a cualquier elemento del diagrama. Estas restricciones serán importantes a la hora de crear el script sql que genere la base de datos.

Para insertar restricciones a los elementos el usuario debe pulsar clic derecho sobre un elemento del diagrama y pulsar "restricciones" dentro del menú desplegable.

Tras ello se abrirá un cuadro de diálogo que permite al usuario crear una lista con las restricciones para ese elemento.

Dominios

Al crear un atributo el usuario debe indicar qué dominio está vinculado con ese atributo. Aunque en un diagrama entidad relación no es necesaria esta información, sí que es importante a la hora de generar el esquema físico ya que todas las columnas de una tabla deben tener un dominio.

El programa ofrece por defecto once de los dominios más comúnmente usados pero también da la posibilidad al usuario de crear su propios dominios.

- Pulsando clic derecho sobre un espacio vacío en el diagrama y pulsando en la opción "crear dominio".
- Seleccionando la pestaña de dominios y haciendo clic en el botón de "nuevo".

Tras ello se abre un cuadro de diálogo en el que el usuario debe indicar el nombre del dominio, el tipo base, y los valores que puede adoptar separados por comas.

Generación de códigos

El siguiente paso tras el diseño del diagrama entidad relación es la generación de los esquemas lógico o físico.

Para ello el usuario debe pulsar en los botones de generar dispuestos en ambos paneles. De existir errores o advertencias, los paneles informarán de forma clara de qué se trata.

Una vez generados los códigos, el usuario tiene la opción de editarlos para añadir, modificar o eliminar cualquier cuestión. Una vez el usuario esté satisfecho con el código, tiene la posibilidad de exportarlo a un archivo en formato texto, o también en formato sql en el caso del esquema físico.

El programa también da la posibilidad de ejecutar el esquema físico directamente en el gestor de base de datos que haya seleccionado.

7.2. Discusión y conclusiones

La aplicación ha mejorado enormemente el aspecto gráfico y se ha adaptado en la medida de lo posible a los estándares de diseño de hoy en día. Una de las limitaciones que se ha sufrido es la de arrastrar las librerías de Jung [14] y la implementación de la interfaz mediante java swing. Hubiese sido muy interesante haber creado una nueva interfaz usando las opciones que proporciona java FX.

También hubiese sido interesante generar el diagrama usando otra librería distinta a Jung, ya que durante todos estos años han surgido mejores soluciones para la creación de este tipo de diagramas. La utilización de esta librería ha supuesto un reto ya que la modificación de las funcionalidades del diagrama resulta muy compleja y apenas existe documentación al respecto.

Al tratarse de un proyecto individual y al ser el principal objetivo hacer que la aplicación volviese a ser usable se decidió centrar los esfuerzos en ello, manteniendo por tanto las librerías de la anterior versión.

También se ha mejorado la funcionalidad de la aplicación corrigiendo errores, mejorando sobre todo el esquema lógico y añadiendo nuevas funcionalidades a la aplicación.

En lo personal, el proyecto me ha permitido aprender en profundidad cómo es el desarrollo de una aplicación de escritorio, así como aumentar mis conocimientos sobre el lenguaje java y sobre la programación orientada a objetos. Especialmente he apren-

dido a como construir una interfaz de usuario completa y funcional.

7.3. Trabajo futuro

A continuación se exponen algunas propuestas para continuar con el desarrollo de la aplicación.

- Añadir las funcionalidades de deshacer y rehacer en el diseño del diagrama entidad relación.
- Añadir la capacidad de insertar elementos directamente conectados a otros acercando el ratón al borde del mismo.
  - Desde una relación se puede crear una entidad.
  - Desde una entidad se puede crear un atributo.
  - Desde un atributo compuesto se puede crear un subatributo.
- Doble clic en un elemento permite editar el nombre del mismo.
- Insertar pestañas para manejar varios archivos abiertos.
- Añadir la capacidad de arrastrar y soltar con los elementos de la barra lateral en el panel de diseño.
- Añadir un panel de frecuencia y volúmenes con los que se pueda condicionar la creación de las tablas en función del uso que se prevé hacer de ellas.
- Permitir modificar los elementos del panel a través del panel de información.
- Crear dos botones (seleccionar y arrastrar) para cambiar el comportamiento del ratón en el diagrama.
- Añadir la funcionalidad de crear temas personalizados directamente desde dentro de la aplicación.
- Sería interesante hacer una versión web de la herramienta.

Capítulo 8

Results conclusions and future work

8.1. Results

DBCASE allows the process of designing and creating a database in a simple way through a desktop interface, focusing on helping the student in his learning process.

Workspace

The user can modify many aspects of the workspace to make his experience with the application more comfortable.

The user can choose the language, theme and perspective he wants. He can also resize the panels to suit his needs.

Creation of a diagram

The first step is the creation and design of a relationship entity diagram. To do this, one of the two perspectives that shows the design panel must be selected.

To add elements to the diagram, the user has two options.

- By right clicking on the design panel. A drop-down menu will be displayed that allows you to insert any type of item.
- Using the sidebar, clicking on the icons that represent the elements.

When inserting elements, their characteristics can be chosen through different dialog boxes. For example, in the case of inserting a new entity, the name must be indicated,

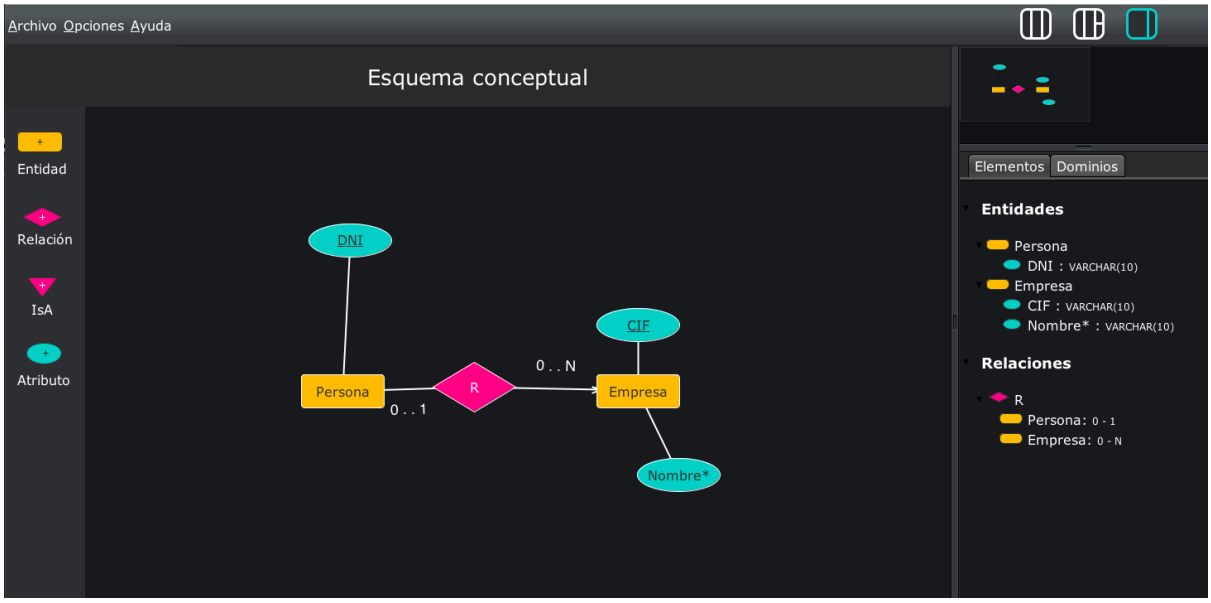


Figura 8.1: Example of custom workspace

also you can indicate the entity is weak respecting to another existing one in the diagram. In that case, you must select that strong entity and name the relationship that unites them.

The procedure to remove elements from the diagram is as follows:

- You must first select an element or a set of elements. (either covering all the elements with the mouse or selecting one by one and holding down the shift key).
- After making the selection you can delete the elements in two ways:
  - Right click on the item, after which a drop-down menu will be displayed, in which one of the options is to remove the item.
  - Pressing the delete key.

  Before the end of the process, a confirmation box will be displayed.

Relations

One of the basic elements in the entity relationship diagrams are the relationships between entities.

To connect a relationship with an entity, the user must right-click on the relationship and click on the "add an entity" option from the drop-down menu. After that, a

dialog box will open in which the entity must be selected and the cardinality chosen to establish the link. It is also possible to indicate a name for the role of the link.

Restrictions

The user can add restrictions to any element of the diagram. These restrictions will be important when creating the sql script that generates the database.

To insert restrictions to the elements, the user must right-click on an element of the diagram and press "restrictions" in the drop-down menu.

After that, a dialog box will open allowing the user to create a list with the restrictions for that element.

Domains

When creating an attribute the user must indicate which domain is linked to that attribute. Although in a relationship entity diagram this information is not necessary, it is important when generating the physical scheme since all the columns of a table must have a domain.

The program offers eleven of the most commonly used domains by default but also gives the user the possibility to create their own domains.

- By right clicking on an empty space in the diagram and clicking on the "create domain" option.
- Selecting the domains tab and clicking on the "new" button.

After that, a dialog box opens in which the user must indicate the domain name, the base type, and the values that can be adopted separated by commas.

Code Generation

The next step after the design of the entity relationship diagram is the generation of the logical or physical schemas.

To do this, the user must click on the generate buttons that appear on both panels. If there are errors or warnings, panels will clearly inform about them.

Once the codes have been generated, the user has the option to edit them to add, modify or delete any question. Once the user is satisfied with the code, he has the possibility of exporting it to a file in text format, or also in sql format in the case of the physical scheme.

The program also gives the possibility to execute the physical scheme directly on the database manager that you have selected.

8.2. Discussion and conclusions

The application has greatly improved the graphic user interface and adapted as far as possible to the design standards of today. One of the limitations that has been suffered is having to use the Jung [14] and java swing libraries. It would have been interesting to have created a new interface using the options provided by Java FX.

It would also have been interesting to generate the diagram using another library other than Jung, since during all these years better solutions for the creation of such diagrams have emerged. The use of this library has been a challenge since the modification of the functionalities of the diagram is very complex and there is hardly any documentation about it.

Being an individual project and being the main objective to make the application usable again, it was decided to focus efforts on it, thus maintaining the libraries of the previous version.

The functionality of the application has also been improved by correcting errors, especially improving the logical schema and adding new functionalities to the application.

Personally, the project has allowed me to learn in depth how is the development of a desktop application, as well as increase my knowledge about java language and object-oriented programming. I especially learned how to build a complete and functional user interface.

8.3. Future work

Below I enumerate some proposals to continue with the development of the application.

- Add the undo and redo functionalities in the entity relationship diagram design.
- Add the ability to insert elements directly connected to others by bringing the mouse to the edge of it.
  - An entity can be created from a relationship.
  - An attribute can be created from an entity.
  - A subattribute can be created from a composite attribute.
- Double clicking on an element allows you to edit its name.
- Insert tabs to handle several open files.
- Add the ability to drag and drop from the sidebar elements in the design panel.
- Add a frequency and volume panel with which you can condition the creation of the tables depending on the use that is expected to make of them.
- Allow to modify the panel elements through the information panel.
- Create two buttons (select and drag) to change the behavior of the mouse in the diagram.
- Add the functionality of creating custom themes directly from within the application.
- It would be interesting to make a web version of the tool.

Bibliografía

Enlaces y recursos útiles mencionados a lo largo de la memoria.

[1] Draw.io
https://www.draw.io/

[2] Smart Draw
https://www.smartdraw.com

[3] EDRPlus
https://erdplus.com

[4] SqlDBM
https://sqldbm.com/Home/

[5] Nimbus Look and Feel
https://docs.oracle.com/javase/tutorial/uiswing/lookandfeel/nimbus.html

[6] Json
https://es.wikipedia.org/wiki/JSON

[7] Singleton
https://es.wikipedia.org/wiki/Singleton

[8] JSplitPane
https://docs.oracle.com/javase/tutorial/uiswing/components/splitpane.html

[9] Graphics2D
https://docs.oracle.com/javase/7/docs/api/java/awt/Graphics2D.html

[10] Math
https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html

[11] Menús en Java Swing
https://docs.oracle.com/javase/tutorial/uiswing/components/menu.html

[12] Integridad referencial
https://es.wikipedia.org/wiki/Integridad_referencial

[13] JUNG Visualization System
http://jung.sourceforge.net/doc/JUNGVisualizationGuide.html

[14] Jung
https://en.wikipedia.org/wiki/JUNG

[15] Modelo Vista Controlador
https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%
93controlador

[16] CVS
https://es.wikipedia.org/wiki/CVS

[17] Repositorio público
https://github.com/miguelarriba/DBCase

[18] Apuntes de Bases de Datos
Javier Arrollo, Yolanda García, Virginia Francisco Gilmartín, Iván Martínez, Fernando Sáenz

Sobre TEFLON

Teflon(cc0 1.0(documentación) MIT(código))es una plantilla de LaTeX creada por David Pacios Izquierdo con fecha de Enero de 2018. Con atribuciones de uso CC0.

Esta plantilla fue desarrollada para facilitar la creación de documentación profesional para Trabajos de Fin de Grado o Trabajos de Fin de Máster. La versión usada es la 1.3.

V:1.3 Overleaf V2 with pdfLaTeX, margin 1in, NO-bib

Contacto

Autor: David Pacios Izquiero

Correo: dpacios@ucm.es

ASCII: asciifdi@gmail.com

Despacho 110 - Facultad de Informática